MIIC-2 Algorithm Requirements

ALGORITHM REQUIREMENTS

PRINCIPLE #1: All MIIC client and server-side algorithms must be generalized to support more than one instrument data type.

PRINCIPLE #2: All MIIC modes of operations support *intercomparison*. The only difference between intercalibration and intercomparison is the level of data used in the analysis. Intercalibration is typically limited to L1 data; intercomparison may use any level data, typically L2. Data mining is the process of extracting select parameters from one or more instrument datasets over a specified geographic region for comparative analyses. L3 OSSE data will accessed using MIIC services and compared to empirical datasets offline. It should be possible to design a generic set of classes and functions to implement all services at all levels of the N-tier system.

MIIC 2 ALGORITHM REQUIREMENTS

1. Extend software to support all modes of operations

- Support LEO-LEO Intercomparison, LEO-GEO Intercomparison, Data Mining, and OSSE Access modes
- Generalize/extend ICPlan class
    - Unique mode data persisted to ICPLan XML output product
    - Execute method runs each plan type
- Integrate code into one MVC controller?

2. Extend application tier analysis software

- Support filtering within application tier
- Support JAIDA-like analysis (TBD)
- Define analysis requirements for each event type (2DHistoGeo, 2DHistoGen, N-Tuple
- Generalize SSF command generation to not limit functionality of SSFs
    - Select any parameter and range for SSF filtering
    - Select any parameter for 2DHisto X and Y axis

3. 2D Histogram server-side function

- *May be complete – if not, update requirements here*
- Verify that any parameter can be assigned to X & Y axes
- Verify that any parameter and range can be selected for filtering
- Verify that the memory allocation/de-allocation problem has been optimized

4. N-Tuple server-side function

- Select any parameters contained in data product to include in N-Tuple output
- Select any parameters and specify ranges for filtering
- Provide data mutator functions that operate on fields contained in each observation record - currently only 2 mutator functions
    - Implement spectral convolution mutator
        - Pass in multiple RSR arrays, eg., MODIS RSRs for bands 20-36; for each band output spectral convolution scalar result (radiance or reflectance)
        - Ref. Tobin [1] downloaded Aqua MODIS IR SRFs from http://www.ssec.wisc.edu/~paulv/Fortran90/Instrument_Information/SRF/SRF_Data.html; use *modisD01_aqua.srf.nc* or *modisD10_aqua.srf.nc*?
    - Implement spectral resampling mutator
        - Pass in reference spectrum (wavelength or wavenumber) and gaussian function
        - Output resampled spectrum

5. Extend and generalize sever-side spatial convolution algorithm (TBD, nice to have but non-trivial)

- In MIIC-1 we demonstrated spatial convolution of MODIS 1 km pixels onto 80 km SCIAMACHY footprints as part of our LEO-LEO use case; we used the SCIAMACHY lat-lon boundary positions provided with each footprint; this may be atypical and therefore not a good candidate for generalization
- Define another technique to convolve high resolution pixels onto lower resolution footprints?
    - Use instrument 2D Point Spread Functions provided by instrument teams (CERES uses instrument pointing vector and 2D response model (PSF) in instrument coordinate system); these algorithms are complex and require a lot of instrument scanning information typically found in L1 data products but not L2; this would require a significant amount of work to generate PSF and apply; each instrument may be significantly different - *not realistic for now unless some instrument team could justify the effort*

- Create artificial geometries (eg., circles) in the Earth coordinate system (lon,lat) centered on observation geolocation, eg., Tobin[1]; we may be able to re-use parts of our current spatial convolution algorithm; for each footprint we would need to know some scanning information to adjust the size of the shape we are using to simulate the PSF response; footprints at the end of each scan would need a larger shape than the nadir footprints.

Type Analysis	**Target**	**Reference**	**Event Predictor**	**Parameters**	**Server-side**
LEO-LEO	Aqua CERES	NPP VIIRS	LEO-LEO	Clouds	2DHisto
LEO-LEO	Aqua CERES	NPP VIIRS	LEO-LEO	Clouds	N-Tuple
LEO-LEO	NPP CrIS	Aqua MODIS	LEO-LEO	MODIS RSRs (bands 20-36) to Avg. CrIS spectrum in App. Tier	2DHisto
LEO-LEO	NPP CrIS	Aqua MODIS	LEO-LEO	MODIS RSRs (bands 20-36) to each spectrum in SSF	N-Tuple + SpConv Mutator
LEO-LEO	NPP CrIS	Aqua AIRS	LEO-LEO	Pass in AIRS spectrum for resampling	N-Tuple + SpResample Mutator
LEO-GEO	GOES-13	NPP VIIRS	LEO-GEO	Radiance	2DHisto
LEO-GEO	GOES-13	NPP VIIRS	LEO-GEO	Radiance	N-Tuple
LEO-LEO	NPP VIIRS, CrIS, CriMSS, ATMS	CALIPSO	LEO-LEO	T, H20, Clouds	2DHisto
LEO-LEO	VIIRS, CrIS, CriMSS, ATMS	CALIPSO	LEO-LEO	T, H20, Clouds	N-Tuple
Data Mining	All Instruments	N/A	Surface Site	Spectra, Clouds, Flux, Radiance	2DHisto
Data Mining	All Instruments	N/A	Surface Site	Spectra, Clouds, Flux, Radiance	N-Tuple
OSSE Access	OSSE	N/A	Grid Cells	Spectra	N-Tuple + SpResample Mutator

Table 1. Example MIIC-2 Test cases; note spectral resampling and spectral convolution N-Tuple mutators within server-side functions.